SUPPLEMENTARY DATA

PostgreSQL was used to get exploratory information on the datasets.

Since the data available in the train table was skewed to show only information on every second day in April and May, two different approaches/models were used to perform the analysis on the dataset to observe the possible differences/similarities in results.

Model A – data analysis using all the dates as presented in the dataset.
Model B – data analysis with every second day withheld in January – March, as in April and May.

Monthly trend
Model A

```
WITH mth AS (SELECT DATE_PART ('month', start_time) AS month_number,
                    COUNT (*) AS number_of_rental
                    FROM train
                    GROUP BY 1
                    ORDER BY 1)
SELECT CASE
        WHEN month_number = 1 THEN 'January'
        WHEN month_number = 2 THEN 'February'
        WHEN month_number = 3 THEN 'March'
        WHEN month_number = 4 THEN 'April'
        ELSE 'May'
        END AS months,
        number_of_rental
FROM mth
```

	months text	number_of_rental bigint
1	January	14714
2	February	17217
3	March	33540
4	April	31697
5	May	38316

Model B

```
WITH mth AS (SELECT DATE_PART ('month', start_time) AS month_number,
        DATE_PART ('day', start_time) AS day_number,
                CASE
                WHEN DATE_PART ('day', start_time)
                IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                THEN 1 ELSE 0 END AS shown_days,
                CASE
                WHEN DATE_PART ('day', start_time)
                IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                THEN 0 ELSE 1 END AS withheld_days,
            COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2,3,4
ORDER BY 1)
SELECT CASE
        WHEN month_number = 1 THEN 'January'
        WHEN month_number = 2 THEN 'February'
        WHEN month_number = 3 THEN 'March'
        WHEN month_number = 4 THEN 'April'
        ELSE 'May'
        END AS months,
        SUM (number_of_rental) as rental_on_alternating_days
FROM mth
WHERE shown_days = 1
GROUP BY 1
ORDER BY 2
```

	months text	rental_on_alternating_days numeric
1	January	7350
2	February	8712
3	March	17736
4	April	31697
5	May	38316

Information about the various timeframes (monthly, day of the week, hour, 30-minute interval)

Model A

```
WITH t AS (SELECT DATE_PART ('month', start_time) AS month_number,
                                    DATE_PART ('dow', start_time) AS day_of_week,
                  DATE_PART ('hour', start_time) AS time_of_day_in_hours,
                        CONCAT (DATE_PART ('hour', start_time), ':',
                    CASE WHEN
                 DATE_PART ('minute', start_time) < 30 THEN 0 ELSE 30 END) AS HalfHour,
                        DATE_PART ('day', start_time) AS day_number,
           COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2,3,4,5
ORDER BY 1)
SELECT  CASE
           WHEN month_number = 1 THEN 'January'
           WHEN month_number = 2 THEN 'February'
           WHEN month_number = 3 THEN 'March'
           WHEN month_number = 4 THEN 'April'
           ELSE 'May'
           END AS months,
                 CASE
           WHEN day_of_week = 0 THEN 'Sunday'
           WHEN day_of_week = 1 THEN 'Monday'
           WHEN day_of_week = 2 THEN 'Tuesday'
           WHEN day_of_week = 3 THEN 'Wednesday'
           WHEN day_of_week = 4 THEN 'Thursday'
           WHEN day_of_week = 5 THEN 'Friday'
           ELSE 'Saturday'
           END AS days_of_the_week,
                 time_of_day_in_hours,
                 HalfHour,
            SUM (number_of_rental) as rental_on_alternating_days
FROM t
GROUP BY 1,2,3,4
ORDER BY 1
```

Model B

```
WITH t AS (SELECT DATE_PART ('month', start_time) AS month_number,
                                    DATE_PART ('dow', start_time) AS day_of_week,
                  DATE_PART ('hour', start_time) AS time_of_day_in_hours,
                        CONCAT (DATE_PART ('hour', start_time), ':',
                   CASE WHEN
                 DATE_PART ('minute', start_time) < 30 THEN 0 ELSE 30 END) AS HalfHour,
                        DATE_PART ('day', start_time) AS day_number,
                        CASE
                          WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 1 ELSE 0 END AS shown_days,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 0 ELSE 1 END AS withheld_days,
            COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2,3,4,5,6
ORDER BY 1)
SELECT  CASE
            WHEN month_number = 1 THEN 'January'
            WHEN month_number = 2 THEN 'February'
            WHEN month_number = 3 THEN 'March'
            WHEN month_number = 4 THEN 'April'
            ELSE 'May'
            END AS months,
                  CASE
            WHEN day_of_week = 0 THEN 'Sunday'
            WHEN day_of_week = 1 THEN 'Monday'
            WHEN day_of_week = 2 THEN 'Tuesday'
            WHEN day_of_week = 3 THEN 'Wednesday'
            WHEN day_of_week = 4 THEN 'Thursday'
            WHEN day_of_week = 5 THEN 'Friday'
            ELSE 'Saturday'
            END AS days_of_the_week,
                  time_of_day_in_hours,
                  HalfHour,
             SUM (number_of_rental) as rental_on_alternating_days
FROM t
WHERE shown_days = 1
GROUP BY 1,2,3,4
ORDER BY 1
```

Maximum and average rental time

Model A

```
SELECT MAX (DATE_PART ('minute', AGE (end_time, start_time))) AS maximum_rental_time_in_min,
        ROUND (AVG (DATE_PART ('minute', AGE (end_time, start_time))) :: NUMERIC, 2) AS average_rental_time_in_min
FROM train
```

maximum_rental_time_in_min double precision	average_rental_time_in_min numeric
59	12.43

Model B

```
WITH t_train AS (WITH mth AS (SELECT bicycle_id AS id,
                        start_time AS start_of_rent,
                        end_time AS end_of_rent,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 1 ELSE 0 END AS shown_days,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 0 ELSE 1 END AS withheld_days,
                start_location AS start_station,
                        end_location AS end_station
                        FROM train)
SELECT id,
        start_of_rent,
        end_of_rent,
        start_station,
        end_station
FROM mth
WHERE shown_days = 1)

SELECT MAX (DATE_PART ('minute', AGE (end_of_rent, start_of_rent))) AS maximum_rental_time_in_min,
        ROUND (AVG (DATE_PART ('minute', AGE (end_of_rent, start_of_rent))) :: NUMERIC, 2) AS average_rental_time_in_min
FROM t_train
```

maximum_rental_time_in_min double precision	average_rental_time_in_min numeric
59	12.64

Station information – top 3 starting points (stations) for bike rental
Model A

```
WITH CTE as (SELECT start_location AS station_id,
                    COUNT (start_location) AS count_of_bicycel_rental
FROM train
GROUP BY 1
ORDER BY 2 DESC)
SELECT DISTINCT
          place_name AS station_name,
          start_of_bicycel_rental
FROM CTE
JOIN station s
ON cte.station_id = s.place_id
ORDER BY 2 DESC
LIMIT 3
```

station_name character varying (250)	count_of_bicycel_rental bigint
Margitsziget	3997
Kálvin tér	3832
Erzsébet tér	3659

Station information – top 3 end points (stations) following a bike rental
Model A

```
WITH CTE as (SELECT end_location AS station_id,
                    COUNT (end_location) AS count_of_bicycel_rental
FROM train
GROUP BY 1
ORDER BY 2 DESC)
SELECT DISTINCT
          place_name AS station_name,
          count_of_bicycel_rental
FROM CTE
JOIN station s
ON cte.station_id = s.place_id
ORDER BY 2 DESC
LIMIT 3
```

station_name character varying (250)	count_of_bicycel_rental bigint
Margitsziget	3999
Kálvin tér	3982
Erzsébet tér	3456

Station information – top 3 starting points (stations) for bike rental
Model B

```
WITH t2 AS (WITH CTE as (SELECT start_location AS station_id,
                        DATE_PART ('month', start_time) AS month_number,
                DATE_PART ('day', start_time) AS day_number,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 1 ELSE 0 END AS shown_days,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                       IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 0 ELSE 1 END AS withheld_days,
                  COUNT (start_location) AS start_of_bicycel_rental
FROM train
GROUP BY 1,2,3,4,5
ORDER BY 6 DESC)
SELECT DISTINCT station_id AS start_station,
            SUM (start_of_bicycel_rental) AS count_of_bicycle_rented_at_station
FROM CTE
WHERE shown_days = 1
GROUP BY 1
ORDER BY 2 DESC)
SELECT DISTINCT
      place_name AS docking_station_name,
            count_of_bicycle_rented_at_station
FROM t2
JOIN station s
ON t2.start_station = s.place_id
ORDER BY 2 DESC
LIMIT 3
```

docking_station_name character varying (250)	count_of_bicycle_rented_at_station numeric
Margitsziget	3336
Kálvin tér	2904
Erzsébet tér	2882

Station information – top 3 end points (stations) following a bike rental
Model B

```
WITH t2 AS (WITH CTE as (SELECT end_location AS station_id,
                        DATE_PART ('month', start_time) AS month_number,
                DATE_PART ('day', start_time) AS day_number,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 1 ELSE 0 END AS shown_days,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                       IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 0 ELSE 1 END AS withheld_days,
                  COUNT (end_location) AS end_of_bicycel_rental
FROM train
GROUP BY 1,2,3,4,5
ORDER BY 6 DESC)
SELECT DISTINCT station_id AS end_station,
          SUM (end_of_bicycel_rental) AS count_of_bicycle_trips_ended_at_station
FROM CTE
WHERE shown_days = 1
GROUP BY 1
ORDER BY 2 DESC)

SELECT DISTINCT
     place_name AS docking_station_name,
          count_of_bicycle_trips_ended_at_station
FROM t2
JOIN station s
ON t2.end_station = s.place_id
ORDER BY 2 DESC
LIMIT 3
```

docking_station_name character varying (250)	count_of_bicycle_trips_ended_at_station numeric
Margitsziget	3333
Kálvin tér	3085
Erzsébet tér	2751

INSIGHTS

Utilizing the information in the complete dataset (Model A) yielded a skewed information about the number of bikes rented in the various months, with the results for January to March showing about two times greater than the value when the data was adjusted to capture the months uniformly (Model B). It is reasonable to estimate the actual rental values for April and May to be around 63,000 and 76,000 respectively, assuming all the days were available for analysis.

Both models were used to analyze the renting patterns across other timeframes, such as daily, days of the week, hourly and, also 30-min intervals. Peak time for bike rentals every day was 17:00 in both models.

The maximum duration of bike rental in both models was 59 minutes, with an overall average rental time of 12.43 minutes and 12.64 minutes for model A and model B respectively.

Margitsziget, Kálvin tér and Erzsébet tér emerged as the top-ranking stations considering stations where most bike rentals started from and subsequent rankings for the destination of these stations when analyzed by both models.

Since both models yielded comparable results, additional exploratory analysis on the stations, bike rental route, and effect of weather on rental count was conducted using model B to ensure uniformity and integrity of the data.

Additional weather information was provided using model A because the weather dataset contained information for every day of January to May. As also observed, the average weather conditions were similar in both models and thus the insights provided are valid.

FURTHER EXPLORATORY ANALYSIS USING MODEL B (WITHHOLDING EVERY SECOND DAY)

Trend from days of the week

```
WITH dow AS (SELECT DATE_PART ('dow', start_time) AS day_of_week,
                    DATE_PART ('day', start_time) AS day_number,
                    CASE
                    WHEN DATE_PART ('day', start_time)
                    IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                    THEN 1 ELSE 0 END AS shown_days,
                    CASE
                    WHEN DATE_PART ('day', start_time)
                    IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                    THEN 0 ELSE 1 END AS withheld_days,
          COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2,3,4
ORDER BY 1)
SELECT CASE
          WHEN day_of_week = 0 THEN 'Sunday'
          WHEN day_of_week = 1 THEN 'Monday'
          WHEN day_of_week = 2 THEN 'Tuesday'
          WHEN day_of_week = 3 THEN 'Wednesday'
          WHEN day_of_week = 4 THEN 'Thursday'
          WHEN day_of_week = 5 THEN 'Friday'
          ELSE 'Saturday'
          END AS days_of_the_week,
           SUM (number_of_rental) as rental_on_alternating_days
FROM dow
WHERE shown_days = 1
GROUP BY 1
ORDER BY 2 DESC
```

	days_of_the_week text	rental_on_alternating_days numeric
1	Friday	16889
2	Saturday	15816
3	Wednesday	15733
4	Tuesday	15243
5	Thursday	13816
6	Sunday	13349
7	Monday	12965

Most rentals were done on a Friday and then Saturday.

Hourly and 30 -min trend

```
WITH t AS (SELECT DATE_PART ('hour', start_time) AS time_of_day_in_hours,
                  CONCAT (DATE_PART ('hour', start_time), ':',
              CASE WHEN
             DATE_PART ('minute', start_time) < 30 THEN 0 ELSE 30 END) AS HalfHour,
            DATE_PART ('day', start_time) AS day_number,
                  CASE
                 WHEN DATE_PART ('day', start_time)
                  IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                  THEN 1 ELSE 0 END AS shown_days,
                  CASE
                  WHEN DATE_PART ('day', start_time)
                  IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                  THEN 0 ELSE 1 END AS withheld_days,
        COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2,3,4
ORDER BY 1)

SELECT  time_of_day_in_hours,
              HalfHour,
         SUM (number_of_rental) as rental_on_alternating_days
FROM t
WHERE shown_days = 1
GROUP BY 1,2
ORDER BY 3 DESC
```

INSIGHTS

Bike renting increased monthly with most renting on Friday and then, Saturday (start of the weekend). Irrespective of the day, most bikes were rented between 16:00 – 19:00, with peak renting occurring 17:00 – 18:00 based on the 30-minute and hourly trend.

Top 10 rental bike routes, from start station to end station

```
WITH CTE as (SELECT
                        CONCAT (start_location, ' - ', end_location) AS route,
                        DATE_PART ('month', start_time) AS month_number,
            DATE_PART ('day', start_time) AS day_number,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                                IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29',
'31')
                        THEN 1 ELSE 0 END AS shown_days,
                        CASE
                        WHEN DATE_PART ('day', start_time)
                                IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29',
'31')
                        THEN 0 ELSE 1 END AS withheld_days,
                COUNT (*) AS count_of_bicycel_rental
FROM train
GROUP BY 1,2,3,4,5
ORDER BY 6 DESC)
SELECT DISTINCT route AS bike_route,
        SUM (count_of_bicycel_rental) AS count_of_bicycle_trips_in_route
FROM CTE
WHERE shown_days = 1
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

bike_route text	count_of_bicycle_trips_in_route numeric
1304 - 1304	1022
1301 - 1304	513
1304 - 1301	338
0604 - 0607	332
0607 - 0604	330
0517 - 0517	312
0508 - 0508	276
0602 - 0607	263
0905 - 0905	258
1301 - 1301	257

SQL query used to transfer the required data from the RDBMS to Power BI for visualization and report.

```
WITH mth AS (SELECT bicycle_id AS id,
                    start_time AS start_of_rent,
                    end_time AS end_of_rent,
                    CASE
                    WHEN DATE_PART ('day', start_time)
                    IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                    THEN 1 ELSE 0 END AS shown_days,
                    CASE
                    WHEN DATE_PART ('day', start_time)
                    IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                    THEN 0 ELSE 1 END AS withheld_days,
               start_location AS start_station,
                    end_location AS end_station
                   FROM train)
SELECT id,
            start_of_rent,
            end_of_rent,
            start_station,
            end_station
FROM mth
WHERE shown_days = 1
```

This query was executed on PostgreSQL to remove every second day from January – March, to ensure datasets were similar with April and May. The query output was saved in CSV format, and transformed using INDEX & MATCH function in MS Excel to integrate the docking station names into the train dataset from the station dataset. This combined table was extracted into Power BI for further transformations with Power Query before final analysis and visualization.

WEATHER AND BIKE RENTAL COUNT

Monthly relationship

The rental pattern across each month was analyzed in relation to several weather factors.

```
WITH r AS (WITH mth AS (SELECT EXTRACT(MONTH FROM start_time) AS rental_month,
      EXTRACT(YEAR FROM start_time) AS rental_year,
CASE
WHEN DATE_PART ('day', start_time)
IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
THEN 1 ELSE 0 END AS shown_days,
CASE
WHEN DATE_PART ('day', start_time)
IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
THEN 0 ELSE 1 END AS withheld_days,
     COUNT (*) AS monthly_rental_count
   FROM   train
   GROUP BY 1,2,3,4)
SELECT rental_year,
rental_month,
  CASE
  WHEN rental_month = 1 THEN 'January'
  WHEN rental_month = 2 THEN 'February'
  WHEN rental_month = 3 THEN 'March'
  WHEN rental_month = 4 THEN 'April'
  ELSE 'May'
  END AS months,
  SUM (monthly_rental_count) as bike_rental
FROM mth
WHERE shown_days = 1
GROUP BY 1,2,3
ORDER BY 2)  ,

w AS (SELECT
      EXTRACT(MONTH FROM time) AS weather_month,
      EXTRACT(YEAR FROM time) AS weather_year,
      ROUND (AVG(tempm):: NUMERIC, 2) AS average_temperature,
      ROUND (AVG(hum) :: NUMERIC,2 ) AS average_humidity,
      ROUND (AVG(wspdm) :: NUMERIC, 2) AS average_windspeed,
   ROUND (AVG(pressurem) :: NUMERIC, 2) AS average_pressure,
   COUNT (*) FILTER (WHERE fog) AS foggy,
  COUNT (*) FILTER (WHERE rain) AS rainy,
  COUNT (*) FILTER (WHERE snow) AS snowfall,
  COUNT (*) FILTER (WHERE hail) AS hail,
  COUNT (*) FILTER (WHERE thunder) AS thunderstorm
```

```
   FROM weather
  WHERE wspdm >= 0
   GROUP BY 1,2)
SELECT rental_month, bike_rental,
     average_temperature, average_humidity, average_pressure, average_windspeed,
  foggy, rainy, snowfall, hail, thunderstorm
FROM r
JOIN w
ON r.rental_month = w.weather_month AND r.rental_year = w.weather_year
ORDER BY r.rental_month, r.rental_month;
```

INSIGHTS

An increase in average temperature with concomitant was observed as the year progressed, providing the greatest correlation between temperature and bike renting. A general increase in average humidity was also observed from analyzing the data, with average wind speed and atmospheric pressure similar across the months.

Snowfall was most in January, and possibly further contributed to reduction in bike rentals. Rainfall was also the most in January. However, a amount of rainfall reported in May was significant also, and perhaps a deep-dive into the daily weather patterns might be necessary to provide explanations.

It was observed that the entire dataset (model A) or model (B) did not significantly alter the weather patterns.

Hourly relationship

```
WITH r AS (WITH mth AS (SELECT EXTRACT(hour FROM start_time) AS rental_hour,
                  EXTRACT(YEAR FROM start_time) AS rental_year,
                        CASE  WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 1 ELSE 0 END AS shown_days,
                        CASE WHEN DATE_PART ('day', start_time)
                        IN ('1', '3', '5', '7', '9', '11', '13', '15', '17', '19', '21', '23', '25', '27', '29', '31')
                        THEN 0 ELSE 1 END AS withheld_days,
                    COUNT (*) AS hourly_rental_count
   FROM   train
   GROUP BY 1,2,3,4)
SELECT rental_year,   rental_hour,
          SUM (hourly_rental_count) as bike_rental
FROM mth
WHERE shown_days = 1
GROUP BY 1,2 ORDER BY 2),
w AS (SELECT
      EXTRACT(hour FROM time) AS weather_hour,
      EXTRACT(YEAR FROM time) AS weather_year,
```

```
    ROUND (AVG(tempm) :: NUMERIC,2) AS average_temperature,
    ROUND (AVG(hum) :: NUMERIC,2) AS average_humidity,
        ROUND (AVG (pressurem) :: NUMERIC, 2) AS average_pressure,
    ROUND (AVG(wspdm) :: NUMERIC, 2) AS average_windspeed,
        COUNT (*) FILTER (WHERE fog) AS foggy,
        COUNT (*) FILTER (WHERE rain) AS rainy,
        COUNT (*) FILTER (WHERE snow) AS snowfall,
        COUNT (*) FILTER (WHERE hail) AS hail,
        COUNT (*) FILTER (WHERE thunder) AS thunderstorm
  FROM weather
      WHERE wspdm  >= 0
  GROUP BY 1,2)
SELECT rental_hour, bike_rental,
    average_temperature, average_humidity, average_pressure, average_windspeed,
        foggy, rainy, snowfall, hail, thunderstorm
FROM r
JOIN w
ON r.rental_hour = w.weather_hour AND r.rental_year = w.weather_year
ORDER BY r.rental_hour, r.rental_year;
```

INSIGHT

Between 15:00 – 18:00, corresponding to the hourly period for highest bike rental, the average temperature was the highest, with associated low average humidity during the day

The results from the above query for the monthly and hourly bike rental activity and weather factors were saved in CSV format. The file was subsequently extracted in MS Excel, and used to create correlation matrix to see the effect of weather factors on monthly rental information (See month_weather_correlation.xlsx and hour_weather_correlation.xlsx).

MONTHLY SPECIFIC INFORMATION

After determining the weather's effect on a monthly level, a more in-depth examination was conducted for each month, providing detailed insights into the influence of weather on bike rentals. Firstly, the rental count was estimated based on the day of the week, the day, and the hour. A comprehensive study was then undertaken to evaluate the impact of weather on renting patterns throughout the specified timeframe.

In the SQL query under this section, replacing the **WHERE monthly = 1** with **(2,3,4,5)** provided information for each month.

Day-to-day rental information in each month

```
WITH sub as (SELECT DATE_PART ('month', start_time) AS monthly,
            DATE_PART ('day', start_time) AS day_number,
            COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2
ORDER BY 2)

SELECT day_number,
           number_of_rental
FROM sub
WHERE monthly = 1  --(OR 2,3,4,5)
ORDER BY 2 DESC
```

Day-to-day information on several weather parameters in each month

A. Average of weather parameters

```
WITH sub as (SELECT DATE_PART ('month', time) AS monthly,
            DATE_PART ('day', time) AS day_number,
              tempm,  hum,   wspdm, pressurem
FROM weather)
SELECT day_number,
                  ROUND (AVG (tempm)::NUMERIC, 2) AS average_daily_temp_in_January,
                  ROUND (AVG (hum) :: NUMERIC, 2) AS average_daily_hum_in_January,
          ROUND (AVG (wspdm)::NUMERIC, 2) AS average_daily_windspeed_in_January,
          ROUND (AVG (pressurem)::NUMERIC, 2) AS average_daily_pressure_in_January
FROM sub
WHERE monthly = 1 --(OR 2,3,4,5)
GROUP BY 1
ORDER BY 2 DESC
```

B. Occurrence of weather parameters.

```
SELECT EXTRACT(DAY FROM time) AS day_number,
            CASE
            WHEN EXTRACT(MONTH FROM time) = 1 THEN 'January'
            WHEN EXTRACT(MONTH FROM time) = 2 THEN 'February'
            WHEN EXTRACT(MONTH FROM time) = 3 THEN 'March'
```

```
        WHEN EXTRACT(MONTH FROM time) = 4 THEN 'April'
        ELSE 'May' END AS month_names,
        COUNT (*) FILTER (WHERE fog) AS foggy,
        COUNT (*) FILTER (WHERE rain) AS rainy,
        COUNT (*) FILTER (WHERE snow) AS snowfall,
        COUNT (*) FILTER (WHERE hail) AS hail,
        COUNT (*) FILTER (WHERE thunder) AS thunderstorm
FROM weather
WHERE EXTRACT(MONTH FROM time) = 1 --(OR 2,3,4,5)
GROUP BY 1,2
ORDER BY 1
```

Days of the week rental information in each month

```
WITH sub as (SELECT DATE_PART ('month', start_time) AS monthly,
                    CASE
        WHEN DATE_PART ('dow', start_time) = 0 THEN 'Sunday'
        WHEN DATE_PART ('dow', start_time) = 1 THEN 'Monday'
        WHEN DATE_PART ('dow', start_time) = 2 THEN 'Tuesday'
        WHEN DATE_PART ('dow', start_time) = 3 THEN 'Wednesday'
        WHEN DATE_PART ('dow', start_time) = 4 THEN 'Thursday'
        WHEN DATE_PART ('dow', start_time) = 5 THEN 'Friday'
        ELSE 'Saturday'
        END AS days_of_the_week,
        COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2
ORDER BY 2)
SELECT days_of_the_week,
       number_of_rental
FROM sub
WHERE monthly = 1 --(OR 2,3,4,5)
ORDER BY 2 DESC
```

Days of the week information on several weather parameters in each month

A. Average of weather parameters

```
WITH sub as (SELECT DATE_PART ('month', time) AS monthly,
                    CASE
        WHEN DATE_PART ('dow', time) = 0 THEN 'Sunday'
        WHEN DATE_PART ('dow', time) = 1 THEN 'Monday'
        WHEN DATE_PART ('dow', time) = 2 THEN 'Tuesday'
```

```
        WHEN DATE_PART ('dow', time) = 3 THEN 'Wednesday'
        WHEN DATE_PART ('dow', time) = 4 THEN 'Thursday'
        WHEN DATE_PART ('dow', time) = 5 THEN 'Friday'
        ELSE 'Saturday'
        END AS days_of_the_week,
         tempm,  hum,  wspdm,  pressurem
FROM weather)
SELECT days_of_the_week,
              ROUND (AVG (tempm)::NUMERIC, 2) AS average_daily_temp_in_January,
              ROUND (AVG (hum) :: NUMERIC, 2) AS average_daily_hum_in_January,
      ROUND (AVG (wspdm)::NUMERIC, 2) AS average_daily_windspeed_in_January,
      ROUND (AVG (pressurem)::NUMERIC, 2) AS average_daily_pressure_in_January
FROM sub
WHERE monthly = 1 --(OR 2,3,4,5)
GROUP BY 1
ORDER BY 2 DESC
```

B. Occurrence of weather parameters.

```
SELECT  CASE
        WHEN DATE_PART ('dow', time) = 0 THEN 'Sunday'
        WHEN DATE_PART ('dow', time) = 1 THEN 'Monday'
        WHEN DATE_PART ('dow', time) = 2 THEN 'Tuesday'
        WHEN DATE_PART ('dow', time) = 3 THEN 'Wednesday'
        WHEN DATE_PART ('dow', time) = 4 THEN 'Thursday'
        WHEN DATE_PART ('dow', time) = 5 THEN 'Friday'
        ELSE 'Saturday'
        END AS days_of_the_week,
        COUNT (*) FILTER (WHERE fog) AS foggy,
        COUNT (*) FILTER (WHERE rain) AS rainy,
        COUNT (*) FILTER (WHERE snow) AS snowfall,
        COUNT (*) FILTER (WHERE hail) AS hail,
        COUNT (*) FILTER (WHERE thunder) AS thunderstorm
FROM weather
WHERE EXTRACT(MONTH FROM time) = 1 --(OR 2,3,4,5)
GROUP BY 1
ORDER BY 3 DESC
```

Hourly information on rental patterns in each month

```
WITH sub as (SELECT DATE_PART ('month', start_time) AS monthly,
                    DATE_PART ('hour', start_time) AS time_of_the_day,
                    COUNT (*) AS number_of_rental
FROM train
GROUP BY 1,2
ORDER BY 2)

SELECT time_of_the_day,
       number_of_rental
FROM sub
WHERE monthly = 1 --(OR 2,3,4,5)
ORDER BY 2 DESC
```

Hourly information on several weather parameters in each month

A. Average of weather parameters

```
WITH sub as (SELECT DATE_PART ('month', time) AS monthly,
                    DATE_PART ('hour', time) AS time_of_the_day,
          tempm,  hum,   wspdm,  pressurem
FROM weather)
SELECT time_of_the_day,
              ROUND (AVG (tempm)::NUMERIC, 2) AS average_daily_temp_in_January,
              ROUND (AVG (hum) :: NUMERIC, 2) AS average_daily_hum_in_January,
       ROUND (AVG (wspdm)::NUMERIC, 2) AS average_daily_windspeed_in_January,
       ROUND (AVG (pressurem)::NUMERIC, 2) AS average_daily_pressure_in_January
FROM sub
WHERE monthly = 1 --(OR 2,3,4,5)
GROUP BY 1
ORDER BY 2 DESC
```

B. Occurrence of weather parameters.

```
SELECT DATE_PART ('hour', time) AS time_of_the_day,
          COUNT (*) FILTER (WHERE fog) AS foggy,
          COUNT (*) FILTER (WHERE rain) AS rainy,
          COUNT (*) FILTER (WHERE snow) AS snowfall,
          COUNT (*) FILTER (WHERE hail) AS hail,
          COUNT (*) FILTER (WHERE thunder) AS thunderstorm
FROM weather
WHERE EXTRACT(MONTH FROM time) = 1 --(OR 2,3,4,5)
GROUP BY 1
ORDER BY 4 DESC
```

INSIGHTS

January

January had the lowest bike rental activity among the five months. The lowest average temperature also recorded in January. The warmest day in January was the 10th, and explains why it was a day with one of the highest bike rentals in the month. Given that 24th and 25th experienced the highest snowfall and rainfall respectively in the month, it is unsurprising that both days experienced the lowest bike rental in January. Sundays, marked with substantial snowfall and rainfall, exhibited the lowest bike rental activity. The peak rental period was observed between 16:00-17:00, coinciding with warmer afternoons between 13:00 – 16:00.

February

Data analysis revealed that, with the exception of snowfall, other weather factors did not offer meaningful insights about the rental pattern. The day with the highest snowfall (9th) correlated with low rental activity. Throughout the month, most rentals were recorded between 15:00 – 18:00, which also corresponded to the warmest hours in the month.

March

Days (25th – 28th) with high bike rental in the month of March had significantly higher temperatures, with 26th recording the highest temperature and simultaneously the highest bike rental. Both days with low bike rental activity in March had substantial rainfall through the day. Throughout the month, most rentals were recorded between 15:00 – 18:00. This observation coincided with the warmest hours in the month.

April

On day 5 with the lowest rental activity, the temperature was 5.23 °C, belonging to the range of lowest temperatures in the month. Most rental was done on Saturday. Similar rental activity based on hour timeframe was observed here.

May

The warmest days were 20th and 19th, with low humidity associated, and thus provides reasonable explanations as to why there was a lot of bike rental, 9th (3468) and 19th (3193). The coldest days in the month were on 3rd and 21st, and this correlated with reduction in bike rental, 21st (815) and 3rd (1212). Rainfall was also the most on the 21st.

Most rental was done on Friday. And throughout the month, most rentals were recorded between 15:00 – 18:00, which also corresponded to the warmest hours in the month. Also, while there was rainfall and thunderstorms recorded in May, there was no noticeable amount in this time-period.